



# Thinking Ahead: Future Architectures from Cray

John M Levesque

Director – Cray Supercomputing Center of Excellence

Cray Chief Technology Office



# Building Momentum in 2006



£20mm contract  
(displaced IBM)



\$85mm contract  
(displaced IBM)



\$ undisclosed



\$250mm contract



## Other Contract Wins

- Toyota Auto Body  
**JANUARY 2006**
- The University of Western Australia  
**MARCH 2006**
- The Institute for Plasma Research  
**MARCH 2006**



\$200mm contract



\$52mm contract  
(displaced IBM)



\$ undisclosed

- *Signed over \$600M worth of contracts*
  - *Over \$300M of revenue generating contracts*
  - *Over \$300M of development contracts*

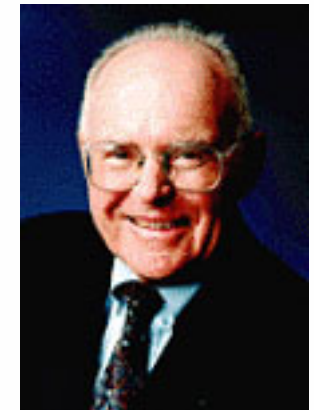
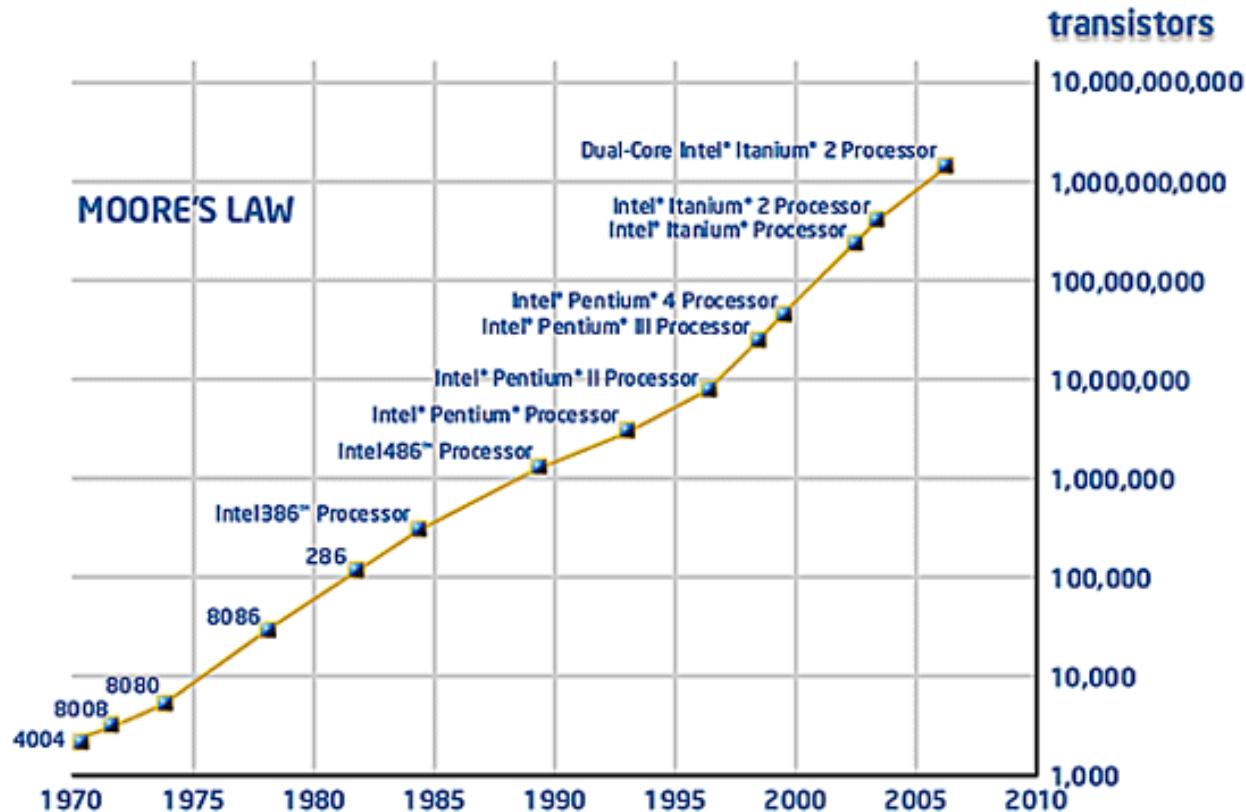
## ***Two Years Later: Cray XT Installs and Orders (over 600 Cabinets sold)***

- Sandia (155 Cabinets, 4 Cabinets)
- ORNL (56 Cabinets, 1 Cabinet, 68 XT4 Cabinets)
- CSCS (18 Cabinets, 2 Single Cabinets)
- PSC (22 Cabinets, 1 Cabinet)
- SS Customers (Total of 11 Cabinets)
- U of Tokyo (1 Cabinet)
- JAIST (4 Cabinets)
- ERDC (44 Cabinets, 1 Cabinet)
- U of Western Australia (3 Cabinets)
- AHPCRC (6 Cabinets)
- AWE (42 Cabinets, 1 Cabinet)
- NERSC (102 Cabinets, 1 Cabinet)
- CSC Finland (12-19 Cabinets)



# Moore's Law

The number of transistors per chip doubles every ~18 months



Gordon Moore, "Cramming More Components Onto Integrated Circuits," *Electronics*, April 19, 1965.



# Processor Performance Increases Slowing

- Moore's Law relates to *density*
- Transistor switching time ~ proportional to gate length
  - feature size X 0.7  $\Rightarrow$  **density X 2, speed X 1.4**
- However, we've also used the extra transistors to....
  1. ... design deeper pipelines (fewer levels of logic per clock)
    - $\Rightarrow$  clock rate has been increasing *faster* than logic transistor speed
  2. ... perform more complicated logic
    - $\Rightarrow$  instructions per clock (IPC) has increased over the years

$\Rightarrow$  ***performance scaled with density for first 30 years of Moore's Law***
- Both of these factors have now ended
  - *Can't pipeline to less than ~6-8 logic levels per clock*
  - *Instructions per clock has settled to about 4 (with less than 4 sustained)*

$\Rightarrow$  ***Increase in per-processor performance has slowed dramatically***

# Another Reason Not to Build Faster Uniprocessors



# Implications for On-chip Architecture

- Okay to overprovision (cheap) flops to maximum utilization of (expensive) memory bandwidth
- Must exploit locality on chip
  - Try to minimize expensive data movement
- Keep voltage and frequency down
  - Concentrate on parallel performance, not single thread performance
- Reduce complexity and overhead
  - Higher fraction of chip doing *computation*, as opposed to control and orchestration
- Commercial response has been to go multi-core
  - Helps alleviate many of these problems, and will likely work well to varying degrees....
  - But raises a number of technical concerns:
    - Rapidly increasing number of processors per system
    - Contention for bandwidth off chip
    - Synchronization, load balancing and managing parallelism across cores
    - Growing memory wall and lack of latency tolerance in conventional processors
    - Still a lot of control overhead in conventional processors

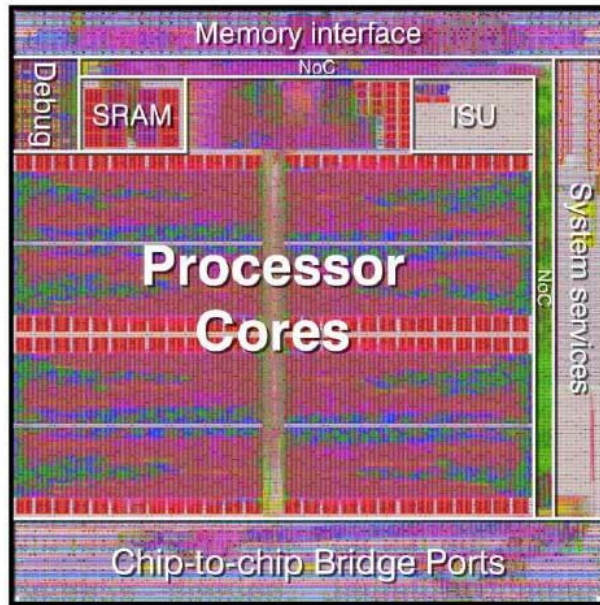
# So, Can We Just Pack Chips with Flops?

- Key is making the system easily programmable
- Must balance peak computational power with generality
  - How easy is it to map high level code onto the machine?
  - How easy is it for computation units to access global data?
- Some examples:
  - XD1 FPGAs
  - Stanford's Streaming Supercomputer project (Merimac)
  - Clearspeed CSX600
  - IBM Cell
- Flop efficiency vs. generality/programmability spectrum:
  - Qualitative only; also influenced by memory system

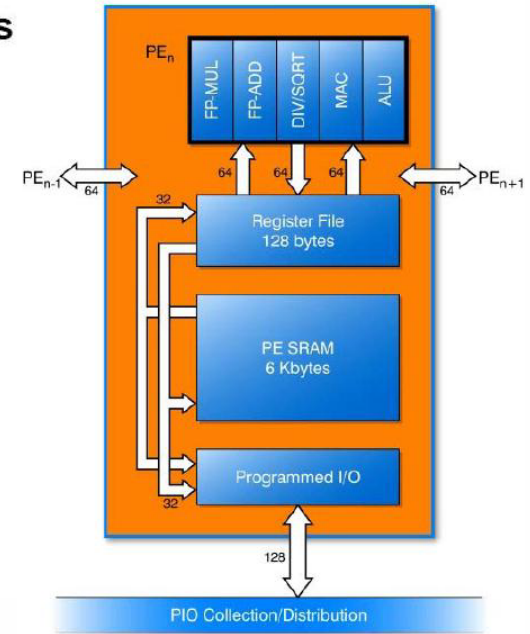




# Clearspeed CSX600

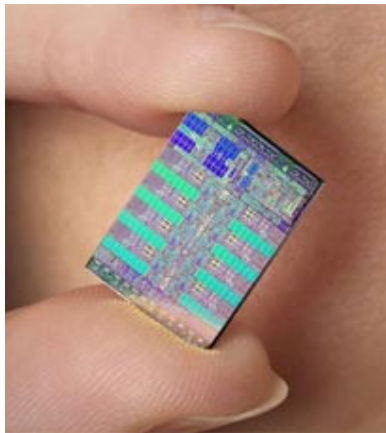


- Array of 96 Processor Elements
- 250 MHz
- IBM 0.13 $\mu$ m FSG process, 8-layer metal (copper)
- 47% logic, 53% memory
  - About 50% of the logic is FPUs
  - Around one quarter of the chip is floating point hardware
- 15 mm x 15 mm die size
- 128 million transistors
- Approx. 10 Watts

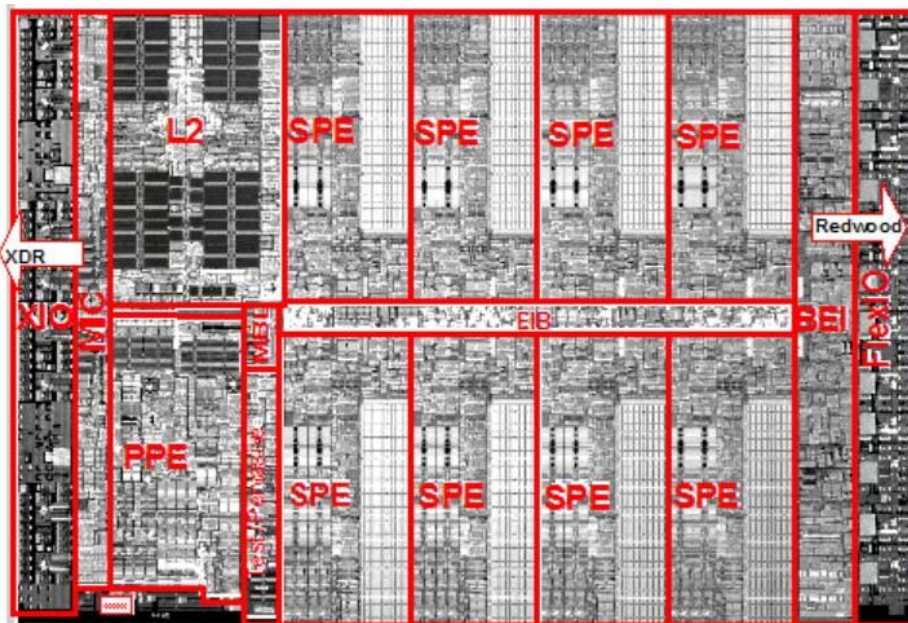


- 50 Gflops on card
- 6 GB/s to on-card local memory (4GB)
- 2+ GB/s to local host memory
- *Doesn't share memory with host*
- *Mostly used for accelerating libraries*
- *No general-purpose compiler available*

# Cell Processor



- Each chip contains:
  - One PowerPC
  - Eight “synergistic processing elements”
- Targeted for:
  - (1) Playstations, (2) HDTVs, (3) computing
- Lots of flops
  - 250 Gflops (32 bit)
  - ~25 Gflops (64 bit)
- 25 GB/s to < 1GB memory
- **Big challenge is programming**
  - SPE's have no virtual memory
  - Can only access data in local 256 KB buffers
  - Requires alignment for good performance
- **No general-purpose compiler available**





# The Cray Roadmap

Pursuing Our Adaptive  
Supercomputing Vision

Scalar

Vector

Multi-threaded



Cray XT3



Cray XD1



Cray X1E



Cray XT4



*BlackWidow*



Cray XMT

*Cray XT4 Upgrade*

Quad Core  
More Flops/Clock

More Cores  
More Flops/Clock

2008

**Phase II: Cascade Program**  
Adaptive Hybrid System

2007

**Phase I: Rainier Program**  
Multiple Processor Types with Integrated Infrastructure and User Environment

2006

**Phase 0: Individually Architected Machines**  
Unique Products Serving Individual Market Needs

*Baker*

*Granite*

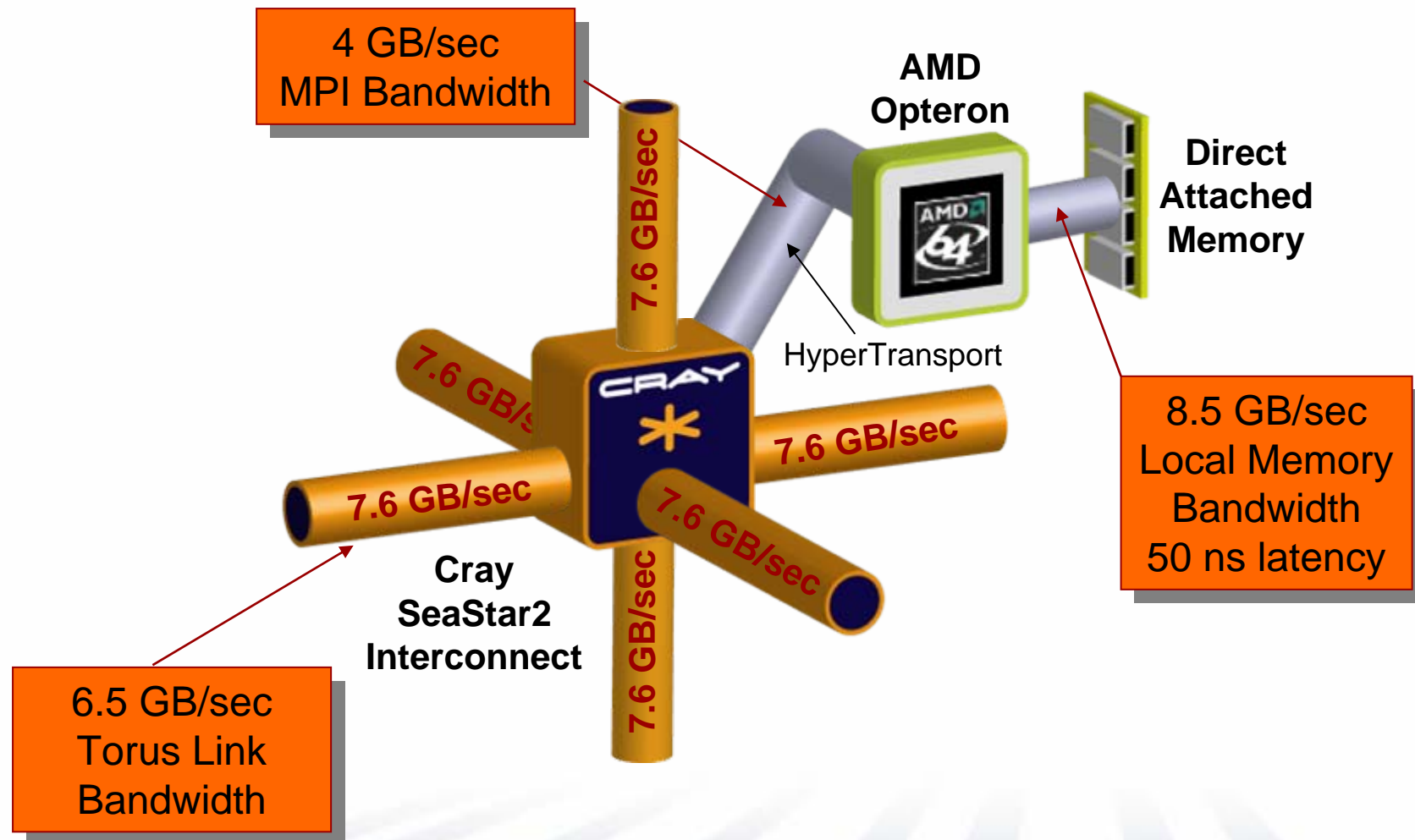
2010

2009

2011

*Marble*

# The Cray XT4 Processing Element: Providing a bandwidth-rich environment



# Rainier Program

- First step toward Adaptive Supercomputing
- Unified User Environment
  - Single login
  - Common service nodes
  - Common global file system
- Multiple processor types using XT4 infrastructure
  - AMD Opteron processors
  - XMT massively multithreaded processors
  - BlackWidow high-bandwidth vector processors
  - FPGA compute blades
- Benefits
  - Leverages technologies across products
  - Reduces overhead for administrators and users
  - Enables hybrid computing





# Cray XMT Platform

## Purpose-Built for Data Analysis and Data Mining

- Architected for large-scale data analysis, not for scientific simulations
  - Exploits thousands of parallel threads accessing large irregular datasets
  - 128 simultaneous threads per processor
  - Scalable to over 8000 sockets and 1M threads
  - Scalable to 128 terabytes of shared memory
- Example Target Markets
  - Government (e.g. pattern matching)
  - Financial services (e.g. fraud detection)
  - Business intelligence (e.g. buying patterns)
  - Healthcare (e.g. genomic based medicine)
  - Digital media (e.g. rendering)
  - Energy (e.g. power management)



# BlackWidow Program

- Project name for Cray's next-generation scalable vector system
- Follow-on to Cray X1E vector system
  - Significantly improved price-performance
  - Special focus on improving scalar performance
- Features:
  - Globally addressable memory
  - Very low overhead synchronization and communication
  - Latency-tolerant processors
  - Massive global bandwidth
  - Highly scalable and configurable
- Integrated with XT infrastructure
  - User login, OS and storage
- Production shipments 2H07



## **CRAY SIGNS \$250 MILLION AGREEMENT WITH DARPA TO DEVELOP BREAKTHROUGH ADAPTIVE SUPERCOMPUTER**

SEATTLE, WA, November 21, 2006 -- Global supercomputer leader Cray Inc. announced today that it has been awarded a \$250 million agreement from the U.S. Defense Advanced Research Projects Agency (DARPA).

Under this agreement, Cray will develop a revolutionary new supercomputer based on the company's Adaptive Supercomputing vision, a phased approach to hybrid computing that integrates a range of processing technologies into a single scalable platform.

[...]

# Motivation for Cascade

## *Why are HPC machines unproductive?*

- Difficult to *write* parallel code (e.g.: MPI)
  - Major burden for computational scientists
- Lack of programming tools to *understand* program behavior
  - Conventional models break with scale and complexity
- Time spent trying to modify code to fit *machine's* characteristics
  - For example, cluster machines have relatively low bandwidth between processors, and can't directly access global memory...
  - As a result, programmers try hard to reduce communication, and have to bundle communication up in messages instead of simply accessing shared memory

*If the machine doesn't match your code's attributes,  
it makes the programming job much more difficult.*

*But codes vary significantly in their requirements,  
so no one machine is best for all codes.*

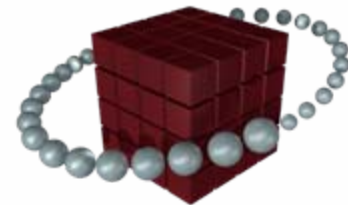
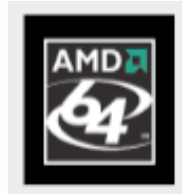
# The Cascade Approach

- Ease the development of parallel codes
    - Legacy programming models: MPI, OpenMP, pthreads
    - Improved programming models: SHMEM, UPC, CAF and Global Arrays
    - New alternative: Global View (Chapel, GMA)
  - Provide programming tools to ease debugging and tuning
  - Design an adaptive, configurable machine that can match the attributes of a wide variety of applications:
    - Fast serial performance
    - SIMD data level parallelism (vectorizable)
    - Fine grained MIMD parallelism (threadable)
    - Regular and sparse bandwidth of varying intensities
- ⇒ Increases performance
- ⇒ Significantly eases programming
- ⇒ Makes the machine much more broadly applicable

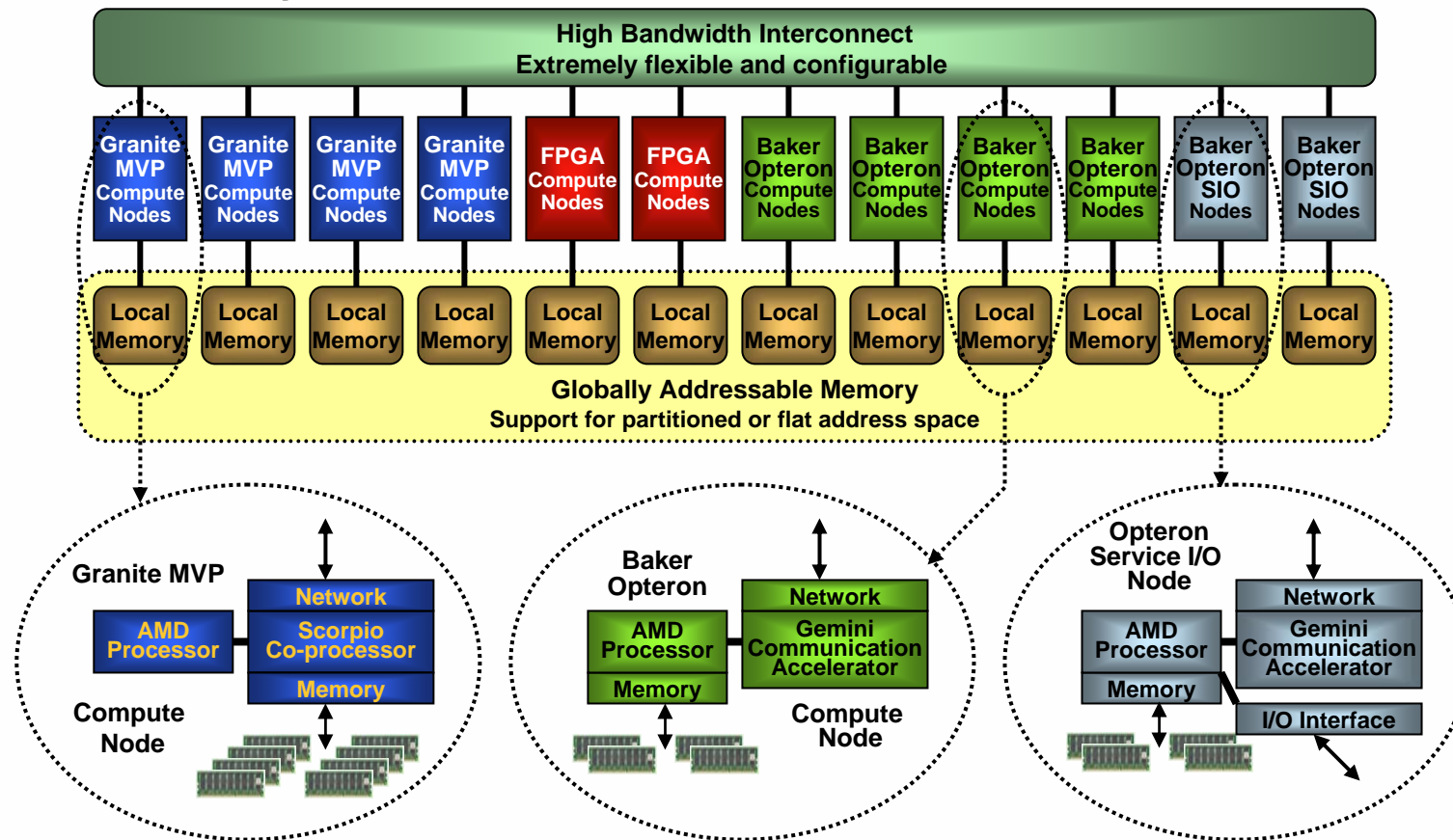


# Cascade Processing Technology

- Build on AMD Opteron™ Processors
  - Industry standard x86/64 architecture
  - Integrated memory controller
    - ⇒ very low memory latency (~50ns)
  - Open high speed interface (HyperTransport)
  - Dual core today with strong roadmap
- Cray *communications* acceleration (Baker and Granite)
  - Support for low latency, low overhead message passing
  - Globally addressable memory
  - Scalable addressing, translation and synchronization
  - Unlimited concurrency for latency tolerance
- Cray *computational* acceleration (Granite)
  - MVP (multi-threaded/vector processing) architecture
  - Exploits compiler-detected parallelism within a node
  - Extremely high single-processor performance



# Cascade System Architecture



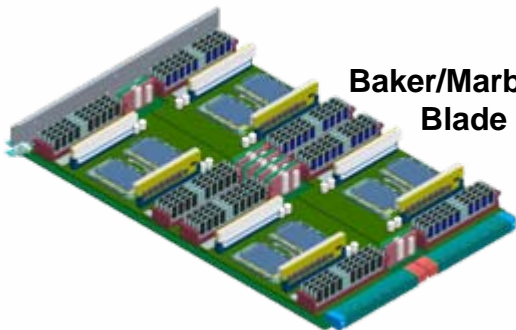
- Globally addressable memory with unified addressing architecture
- Configurable network, memory, processing and I/O
- Heterogeneous processing across node types, and within MVP nodes
- Can adapt at **configuration** time, **compile** time, **run** time

# Highly Extensible System Packaging

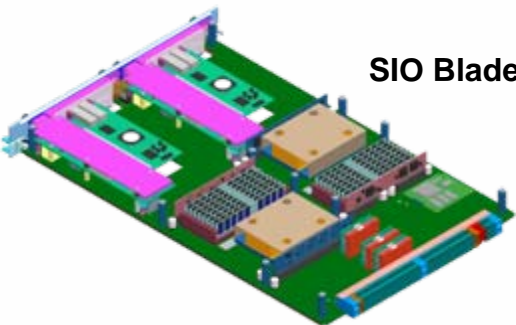


- Will extend XT4 cabinet, shown here
- Improvements in density and cooling
- Multi-year upgrade paths
  - Sockets and DIMMs
  - Compute and network blades

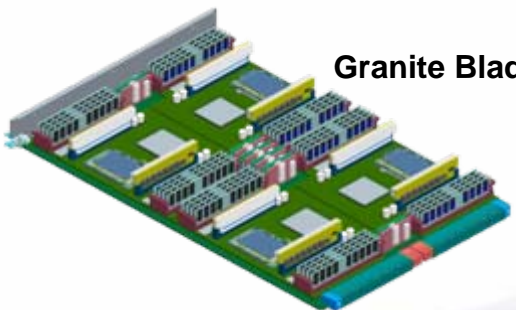
**Baker/Marble  
Blade**



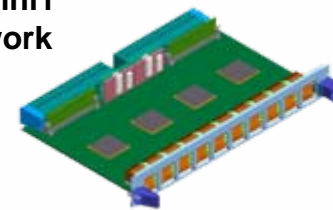
**SIO Blade**



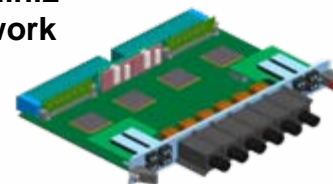
**Granite Blade**



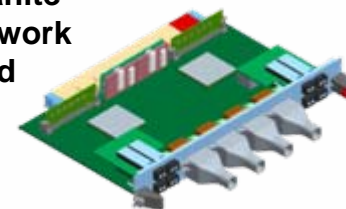
**Gemini1  
network  
card**



**Gemini2  
network  
card**



**Granite  
network  
card**



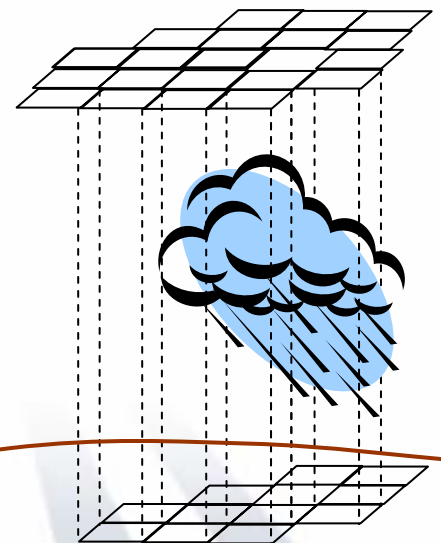
# Uniquely Programmable

- Ease of programming and portability are critical
  - The Granite MVP accelerator is the *only* accelerator that does *not* require changing the programming model!
- Granite will have general purpose C and Fortran compilers
  - Uses same code as written for other machines
  - Automatically splits up code to run on the Opteron and the Scorpio co-processor
  - Automatically vectorizes and multithreads code
  - Chooses the best mode for every loop nest
- User does not have to tune for one style of machine
- Get the benefit of multiple machine types rolled into one
- Globally addressable memory and huge network bandwidth make communication very easy and efficient
- Users can concentrate on the *science*

# Example Application:

## Weather Research & Forecasting (WRF) Model

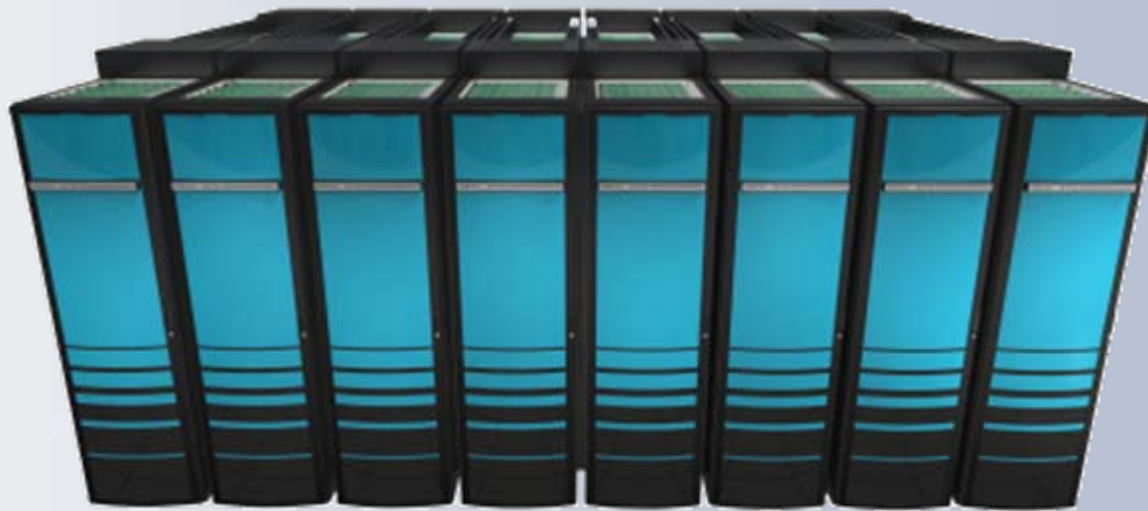
- Mesoscale numerical weather prediction system
  - Regional forecast model (meters to thousands of kilometers)
- Operational forecasting, environmental modeling, & atmospheric research
  - Key commercial application for Cray (both vector & scalar MPP systems)
- Accelerating WRF performance on Cascade:
  - Part of the code is serial:
    - Runs on **Opteron** for best-of-class serial performance
  - Most of the code vectorizes really well
    - Dynamics and radiation physics
      - ⇒ Runs on MVP accelerator in **vector mode**
  - Cloud physics doesn't vectorize
    - Little FP, lots of branching and conditionals
    - Degrades performance on vector systems
    - Vertical columns above grid points are all independent
      - ⇒ Runs on MVP accelerator in **multithreaded mode**





# Supercomputing Innovation from Cray

- Keeping our eye on the foundation of our product line
  - World class scalability – operating systems and system management
  - Strong system balance
  - Next generation Baker system will significantly advance this capability
- Increasing opportunities to exploit heterogeneity
- Cray provides multiple processor types for different types of applications
  - General purpose Opteron processors
  - Massively multithreaded processors
  - High-bandwidth vector processors
  - FPGA accelerators
- Cray's Rainier program ties these processors into a single, unified system
  - Based on the highly scalable XT4 infrastructure
  - Shared global file system and unified user environment
- Cray's Cascade program increases integration and sophistication
  - Common globally-addressable memory
  - MVP processors adapt their operation to fit the code
- Adaptive Supercomputing Vision
  - Optimize performance via high bandwidth and multiple processor types
  - Ease programming by allowing the system to adapt to the application



***Thank You.  
Questions?***

